

# Compressive Change Retrieval for Moving Object Detection

Murase Tomoya

Tanaka Kanji

**Abstract**—Change detection, or anomaly detection, from street-view images acquired by an autonomous robot at multiple different times, is a major problem in robotic mapping and autonomous driving. Formulation as an image comparison task, which operates on a given pair of query and reference images is common to many existing approaches to this problem. Unfortunately, providing relevant reference images is not straightforward. In this paper, we propose a novel formulation for change detection, termed compressive change retrieval, which can operate on a query image and similar reference images retrieved from the web. Compared to previous formulations, there are two sources of difficulty. First, the retrieved reference images may frequently contain non-relevant reference images, because even state-of-the-art place-recognition techniques suffer from retrieval noise. Second, image comparison needs to be conducted in a compressed domain to minimize the storage cost of large collections of street-view images. To address the above issues, we also present a practical change detection algorithm that uses compressed bag-of-words (BoW) image representation as a scalable solution. The results of experiments conducted on a practical change detection task, “moving object detection (MOD),” using the publicly available Malaga dataset validate the effectiveness of the proposed approach.

## I. INTRODUCTION

Change detection, or anomaly detection, from street-view images acquired by an autonomous robot at multiple different times, is a major problem for robotic mapping and autonomous driving. Given a visual image of the robot’s surroundings as a query, the goal of change detection is to search over a database or a collection of previously acquired images to identify regions that correspond to environment changes (e.g., appearance of new objects), which comprise the “change mask” [1]. A key issue is that the change mask should not contain unimportant or nuisance forms of change, such as those induced by difference in views and sensor noises. In this sense, change detection is similar in its objectives to anomaly detection [2], in which the goal is to detect anomalies that are interesting to the observer.

The problem of change detection from street-view images has drawn much research attention over the past decade [3]–[5], and has resulted in the production of many interesting and effective algorithms. The use of 3D line segments for change detection, in which line segments are matched between multiple views, 3D line segments reconstructed, and 3D images then compared to detect changes, is proposed in [3]. Change detection from Google StreetView image panoramas acquired by moving vehicles, in which a coarse



Fig. 1. Compressive change retrieval for moving object detection. In our change detection formulation, the robot’s current view image is compared against reference images that are retrieved from street-view images using the view image as query. Shown in the figure is an example of change detection result: detected anomaly (small colored points) and anomaly score (color bar).

3D geometry of the scene is recovered and then registered with previously acquired reference images of the location, and further semantic content of current and previous views are exploited to gather additional evidence about the change hypothesis, is proposed in [4]. “City-scale” change detection from 3D city models and panoramic images captured by a car driving around the city, in which geometric changes are detected by comparing images and 3D models, under inaccuracies in input geometry, errors in the image’s GPS data, as well as limited amount of information owing to sparse imagery, is proposed in [5].

Formulation as an *image comparison* task, which operates on a given pair of query and reference images, is common to the majority of approaches such as these. Unfortunately, providing relevant reference images is non-trivial and the main topic of ongoing place recognition studies. To date, most state-of-the-art systems simply assume relevant reference images are given, or rely on the availability of a reference image’s viewpoint (e.g., GPS), which limits their application scenarios.

In this paper, we propose a novel formulation for change detection, termed *compressive change retrieval*, which does not require a relevant reference image, but instead can operate on a query image and similar reference images retrieved from the web (Fig. 1). This study is motivated by recent

Our work has been supported in part by JSPS KAKENHI Grant-in-Aid for Young Scientists (B) 23700229, and for Scientific Research (C) 26330297 (“The realization of next-generation, discriminative and succinct SLAM technique: PartSLAM”).

T. Murase and K. Tanaka are with Graduate School of Engineering, University of Fukui, Japan. [tnkknj@u-fukui.ac.jp](mailto:tnkknj@u-fukui.ac.jp)

progress in large-scale image retrieval and publicly available street-view images, thanks to which it is possible to obtain a collection of similar street-view images (i.e., candidate reference images) to a given query image. Compared to previous formulations, there are two sources of difficulty. First, the retrieved reference images may frequently contain non-relevant reference images, because even state-of-the-art image retrieval techniques suffer from retrieval noises. Second, image comparison needs to be conducted in a compressed domain to minimize the storage cost of large collections of street-view images. To address the above issues, we present a practical change detection algorithm that uses compressed bag-of-words (BoW) image representation as a scalable solution. The following contributions are made in this paper: (1) We reformulate the change detection task as compressive change retrieval and thereby extend change detection to the case of multiple noisy reference images. (2) We present a practical change detection algorithm for compressed BoW image representation, spatial analysis, and occlusion reasoning. (3) We implement the change detection framework on a practical application of “moving object detection (MOD)” from street-view images. Experimental results using the publicly available Malaga dataset validate the effectiveness of the proposed approach.

Our approach is orthogonal to most of the existing approaches to MOD. The experimental scenario that we consider for change detection is a practical application of MOD, where the goal is to detect moving objects (e.g., cars) in a single urban image by comparing the query image with similar retrieved images. Existing solutions to moving object detection can be broadly categorized into those using motion cues (e.g., motion segmentation, and moving camera background subtraction) and those based on prior knowledge (e.g., pre-trained object detector, and change detection). The types of prior knowledge used in the latter category can be divided into prior on foreground (i.e., labeled object examples) and prior on background (i.e., image acquired at different times). All of the above approaches are far from perfect, and suffer from image processing noise, offline costs for acquiring prior, or both. Our proposed approach can be viewed as a prior based approach with a novel low cost prior (i.e., random street-view images), which has not been sufficiently explored in the literature and is the main contribution of our study.

## II. APPROACH

For clarity of presentation, we first describe the baseline approach, which is the base for our approach and is a performance comparison benchmark in the experiments described in section III. Then, we present a solution to the change retrieval task as an extension of the baseline system.

### A. Baseline System

The main process consists of three steps (Fig.3): (1) image retrieval, (2) change proposal, and (3) decision. In the first step, the aim is to obtain a set of candidate reference images by retrieving the collection of street-view images.

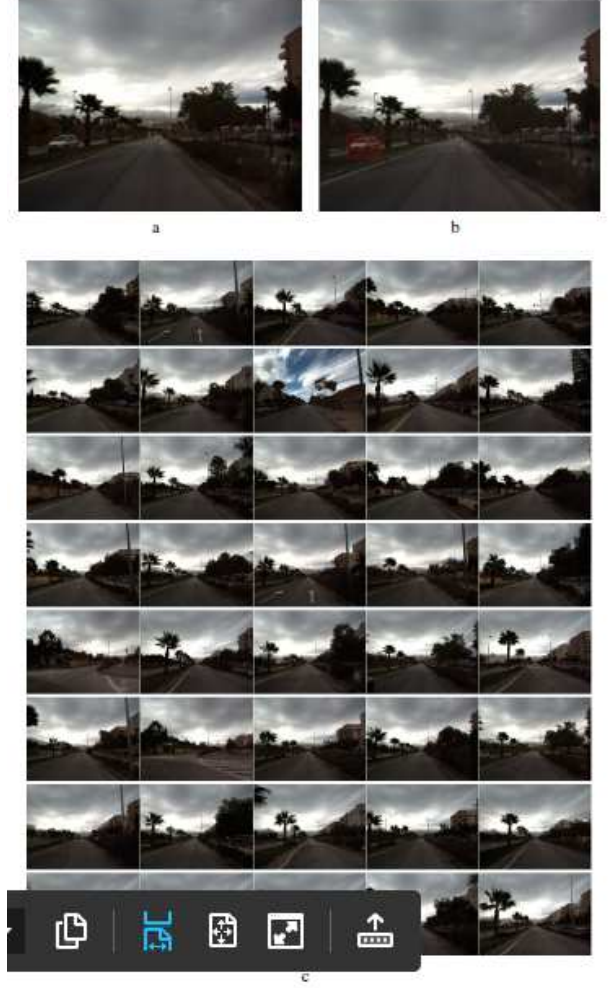


Fig. 2. Image retrieval: (a) query image, (b) ground-truth moving object, and (c) retrieved reference images.

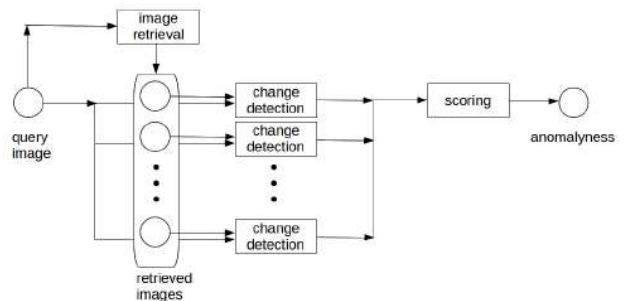


Fig. 3. Main processing.

For image retrieval, any visual features (e.g., SIFT, SURF) can be used but we chose DCNN features from Alexnet [6] because of their excellent performance. Fig. 2 shows an example of the image retrieval process. As can be seen, relevant images are successfully retrieved thanks to the DCNN features. However, it can also be seen that there is a non-negligible number of non-relevant images, due to the retrieval noise.

In the second step, the aim is to compare visual local

features between the query image and each reference image of interest to evaluate the probability of individual features in the query image corresponding to changes. To this end, we employed 128-dim SIFT descriptors at interest keypoints as features, and computed the dissimilarity in terms of L2 norm as anomaly.

The third step integrates the results of scene comparisons for all the reference images and, based on the result, we compute the anomaly of each feature in the query image. The basic idea is to compute the minimum of the dissimilarity (i.e., L2 norm) over all the reference images and view it as the anomaly of the given feature:

$$A(f^q) = \min_{r \in R} \min_{f^r \in r} |f^q - f^r|_2, \quad (1)$$

where  $R$  is the set of reference images, and  $f^q$  and  $f^r$  are features in the query and a reference image  $r$ .

### B. Bag-of-Words Extension

The basic idea of underlying bag-of-words image representation is translation of the visual local features in a given image to an unordered collection of visual words. In preprocessing, we prepare a fine visual vocabulary consisting of 1M exemplar visual features each of which corresponds to a different visual word. To translate a given feature in a reference image, we search over the vocabulary to find the feature's nearest neighbor exemplar, and assign the exemplar's ID as the feature's visual word. The result is an unordered collection of visual words, termed bag-of-words. All the features in all the street-view images are stored in the inverted index of visual words and retrieved therefrom.

Our strategy for dissimilarity evaluation is an instance of asymmetric distance computation (ADC) [7], which only encodes the local features of the reference images, not the local feature in the query image. This is in contrast to symmetric distance computation (SDC) employed by typical BoW systems, which encodes both query and database features. Our ADC-based method runs kNN search over the set intersection between the reference images' features and the vocabulary's features using the given query feature. Then, it computes the L2 distance between the query feature and the nearest neighbor feature as the anomaly of the query feature.

### C. Local Geometry

Alignment is a standard preprocessing step in change detection [1] that aligns local features between query and reference images. This preprocessing enables change detection algorithms to compare only those features that are spatially near to each other, and reduces incidences of similar but spatially distant objects not being detected. A naive strategy is to perform one-to-one registration for every pair of query and reference images. In our case, it is unfortunately impractical to run the alignment step for all the pairs of query and reference images, as the collection of street-view images is large. Instead, we here describe an efficient strategy for the alignment that can operate on the efficient inverted index.

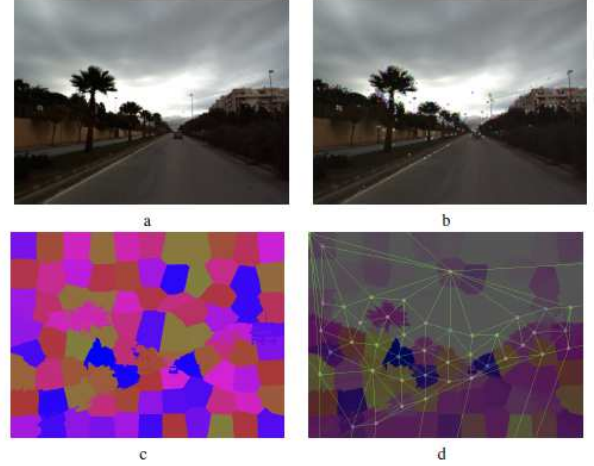


Fig. 4. Image processing. (a) Input image. (b) SIFT keypoints. (c) Superpixel regions. (d) Result of local geometry analysis. The points indicate center points of superpixel regions. The lines connect geometric neighbor points. Only those features inside the visibility region are used.

In preprocessing, we analyze the LG of local feature positions and store the result in the inverted index (Fig.4). To this end, we adapt triangulation-based analysis of LG, which was originally proposed for a different application, specifically, “scalable logo recognition”, in [8]. We capture feature geometry using Delaunay triangulation on local feature positions. Because a Delaunay graph is quite sensitive to the position's noise, we employ superpixel image segmentation in [9] and then each superpixel's center position acts as the position of the features that belong to the superpixel. To this end, we need to obtain and store two kinds of information: (1) the IDs of the superpixels that are adjacent to each superpixel, and (2) the ID of superpixel to which each feature belongs.

The alignment step first establishes correspondence between each feature in the query image and visual words in the reference image to find *strong matches*  $\{1, \dots, K\}$  that satisfy the SIFT ratio condition in [10]:

$$\text{dist}[1] < \dots < \text{dist}[K] < 0.8 * \text{dist}[K+1], \quad (2)$$

where  $d[k]$  is the L2 distance between the query feature of interest and the  $k$ -th nearest neighbor match. If strong matches are found (i.e.,  $K \geq 1$ ), each query feature is compared with those reference features that belong to a selected subset of superpixels (rather than all the features in the reference image), which are adjacent to those superpixels that have at least one strong match.

### D. Visibility Analysis

We also estimate image regions commonly visible from both query and reference images, which enables change detection algorithms to compare only those features that belong to the commonly visible regions. In this study, the visible regions were simply modeled as a pair of bounding boxes  $[x_{min}^i, x_{max}^i] \times [y_{min}^i, y_{max}^i]$  ( $i \in \{q, r\}$ ) in query ( $q$ ) and reference ( $r$ ) images. To determine the bounding box for



either image, we first sort the matched visual features in the image in ascending order of  $x, y$ -coordinate, and define  $0.1 * |M^i|$ -th and  $0.9 * |M^i|$ -th elements ( $\delta = 0.1$ ) in the sorted list as the  $x, y$ -locations of the vertexes of the bounding box. Then, we further enlarge the width and height of the bounding box by a scaling factor of  $1/(1 - \delta)$  to increase robustness. In change detection, we use only those features that belong to superpixels whose center positions lie within the bounding boxes.

### III. EXPERIMENTS

We verified our method on a large collection of urban images and compared the results obtained to the baseline change detection method. The experimental scenario considered for change detection was a practical application of “moving object detection (MOD),” in which the goal is to detect moving objects (e.g., cars) in a given query image by comparing the query image with the retrieved similar images. Such detection is a practical scenario because existing solutions for MOD from motion segmentation, such as moving camera background subtraction, are far from perfect, and even learning-based techniques are ill-posed owing to visual diversity of the appearance of an object. This is also a challenging scenario because the experimental environment contains many similar places and perceptual aliasing makes place recognition a difficult task, and yields retrieval noise. Furthermore, MOD solely from an object’s appearance is difficult because there are many non-moving objects (e.g., parking cars) with quite similar models.

Fig. 5 shows the results of 1-NN BoW matching for each query feature over features in a reference image. As can be seen, even 1-NN matches contain both relevant and irrelevant features, which make our anomaly detection task a challenging one. Fig. 6 shows examples of change detection. It can be seen that the proposed method is successful even when there are many similar non-changed cars within one query image.

#### A. Settings

For the evaluation, we used image sequences from the Malaga dataset [11]. The Malaga dataset contains GPS data, two cameras facing forward in the direction of vehicle motion, as well as LIDAR data. We use the left eye’s images with a resolution of  $1024 \times 768$  for our change detection task. Although our application scenario is monocular visual recognition, we employed the stereo image sequence to collect ground-truth viewpoint information for place recognition and change detection, where stereo SLAM with visual odometry with loop closure constraints is used for estimating trajectory.

We considered a practical place recognition scenario, called loop closing [12], derived from the field of visual SLAM, in which a robot traverses a loop-like trajectory and then returns to the previously explored location. More formally, the relevant image pair is defined by a database image that satisfies two conditions: 1) Its viewpoint from each query’s viewpoint is nearer than other candidates. 2) Its

distance traveled along the robot’s trajectory is sufficiently larger (200 frames) than that of the query image. Owing to condition 2, a relevant pair of images become dissimilar to each other, which makes our place recognition and change detection tasks challenging. To this end, we used datasets #5, #6, #7, #8, and #10 from the Malaga dataset, because they have “loop-closing” situations, each of which consists of 4816, 4618, 2121, 10026, and 17310 images.

We implemented the proposed method in C++. In accordance with [4], we used SIFT [10] as visual features for image comparison and change detection. For the superpixel segmentation, we used SLIC superpixel code [9] with parameters  $nr = 100$  and  $nc = 50$ . In the above settings, the number of SIFT features extracted per image was in the range 0-5551. For bag-of-words translation, we used a visual vocabulary with 1M words. The default number of candidate reference images per query was set at 40. The techniques described in subsections II-C and II-D are respectively termed “local geometry (LG)” and “visibility analysis (VA)” below.

To evaluate different methods under comparison, we used a ranking-based performance measure, for which a smaller value indicates better performance. The Ranking is defined as the rank of the feature that belongs to the ground-truth location of changed objects, in a list of features sorted in descending order of the anomaly score.

We selected 110 images at random as query images for the experiments, such that for each moving object that the camera encountered one or a few query images appeared. We annotated the ground-truth location of changed objects with a form of bounding box in the query image. When there were multiple features in the bounding box, we selected and used the feature with the highest anomaly score for computation of the Ranking.

#### B. Results

We also compared the results obtained by the proposed change detection algorithm with those from the baseline algorithm using non-quantized SIFT features. For fair comparison, both the proposed and the baseline algorithms employ the techniques described in II. Tables I “CCR” and “baseline” show the top- $X\%$  ( $X \in \{1, 3, 5\}$ ) detection performance for the proposed method (CCR) and the baseline method (baseline). It can be seen that the proposed method is comparable to the baseline method despite the fact that the proposed method is based on vector-quantized representation of images, i.e., bag-of-words, which is significantly more compact.

We set the default number of candidate reference images per query as 40. This means that for change detection, each query image was compared against 40 different similar reference images retrieved from the database. We tested several different values for the number of reference images. Table II shows the results of top-3% NR performance for the values, 40, 20, 10, and 5. As expected, the performance of the proposed CCR method was increasingly better as the reference image set became larger and more informative.



Fig. 6. Results of change detection.

We also compared the cases with and without the “Local Geometry (LG)” and “Visibility Analysis (VA)” techniques described in II-C and II-D. In Table II, “CCR,” “CCR+LG,” and “CCR+LG+VA” indicate the results with and without the LG and VA techniques. It can be seen that, compared to the case of the proposed algorithm, in both cases, performance fell. In particular, the performance decrease is significant in the case of CCR without LG. Thus, it is clear that the proposed LG and VA techniques both effectively improve change detection performance in the proposed CCR framework.

#### IV. CONCLUSIONS

In this paper, we addressed the problem of compressive change retrieval for moving object detection (MOD). Unlike existing algorithms for change detection that operate on given reference images, we presented a practical solution that can operate on large collections of unorganized street-view images. Three contributions were made in this paper: (1) We reformulated the change detection task as compressive change retrieval and extended the previous change detection to the case of multiple noisy reference images. (2) We presented a practical change detection algorithm for compressed BoW image representation, spatial analysis, and occlusion reasoning. (3) We implemented the change detection framework on a practical application of MOD. Experimental results using the publicly available Malaga dataset validate the effectiveness of the proposed approach. Our future work will focus on an integrated MOD, in which

the proposed detection algorithm will be combined with other recognition algorithms based on motion cues, background subtraction, and visual learning, in order to further improve its change detection performance.

#### REFERENCES

- [1] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, “Image change detection algorithms: a systematic survey,” *IEEE transactions on image processing*, vol. 14, no. 3, pp. 294–307, 2005.
- [2] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
- [3] I. Eden and D. B. Cooper, “Using 3d line segments for robust and efficient change detection from multiple noisy images,” in *European Conference on Computer Vision*. Springer, 2008, pp. 172–185.
- [4] J. Košečka, “Detecting changes in images of street scenes,” in *Asian Conference on Computer Vision*. Springer, 2012, pp. 590–601.
- [5] A. Taneja, L. Ballan, and M. Pollefeys, “Geometric change detection in urban environments using images,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2193–2206, 2015.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [7] T. Kanji, “Self-localization from images with small overlap,” in *IEEE/RISJ IROS*, 2016.
- [8] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis, “Scalable triangulation-based logo recognition,” in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. ACM, 2011, p. 20.
- [9] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels,” *Tech. Rep.*, 2010.
- [10] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

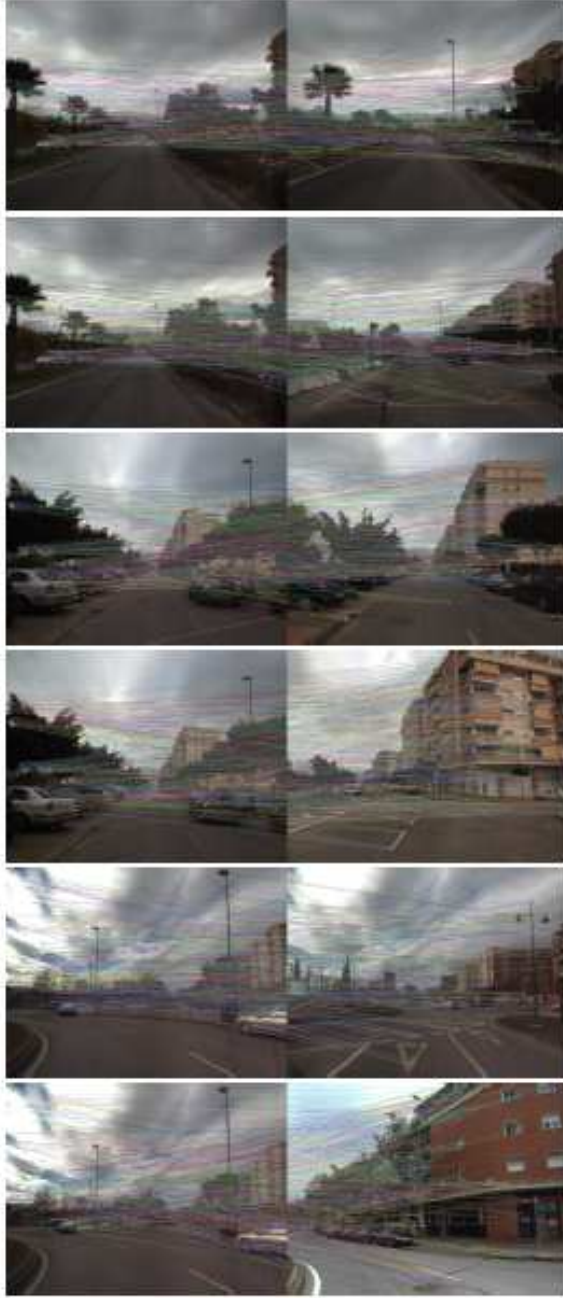


Fig. 5. Matching BoW features.

- [11] J.-L. Blanco, F.-A. Moreno, and J. Gonzalez, "A collection of outdoor robotic datasets with centimeter-accuracy ground truth," *Autonomous Robots*, vol. 27, no. 4, pp. 327–351, 2009.
- [12] H. Shogo and T. Kanji, "Partslam: Unsupervised part-based scene modeling for fast succinct map matching," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1582–1588.

TABLE I  
CHANGE DETECTION PERFORMANCE IN [%]

#ref	algorithm	top-1	top-2	top-5	top-10	top-20	top-50
40	DM	17.6	20.7	<b>39.2</b>	<b>54.7</b>	62.9	81.5
	DM+LG	17.6	20.7	<b>39.2</b>	<b>54.7</b>	62.9	81.5
	CCR	<b>18.6</b>	20.7	31.0	47.5	64.0	81.5
	CCR+LG	13.5	<b>23.8</b>	<b>39.2</b>	51.6	<b>70.2</b>	<b>88.7</b>
	CCR+LG+VA	12.4	22.7	36.1	46.4	67.1	83.6
20	DM	13.5	18.6	29.9	42.3	61.9	81.5
	DM+LG	<b>20.7</b>	<b>24.8</b>	<b>32.0</b>	<b>49.5</b>	65.0	<b>83.6</b>
	CCR	13.5	21.7	<b>32.0</b>	46.4	<b>66.0</b>	80.5
	CCR+LG	12.4	19.6	28.9	44.4	60.9	79.4
	CCR+LG+VA	15.5	22.7	<b>32.0</b>	45.4	60.9	80.5
10	DM	<b>17.6</b>	21.7	27.9	44.4	64.0	80.5
	DM+LG	<b>17.6</b>	<b>23.8</b>	<b>32.0</b>	<b>46.4</b>	60.9	81.5
	CCR	10.4	18.6	31.0	45.4	<b>65.0</b>	<b>82.5</b>
	CCR+LG	11.4	14.5	26.9	40.3	57.8	81.5
	CCR+LG+VA	12.4	15.5	24.8	38.2	55.7	80.5
5	DM	<b>18.6</b>	<b>21.7</b>	31.0	46.4	61.9	76.3
	DM+LG	12.4	19.6	<b>36.1</b>	<b>52.6</b>	62.9	78.4
	CCR	14.5	18.6	34.1	44.4	<b>66.0</b>	<b>81.5</b>
	CCR+LG	15.5	20.7	34.1	45.4	58.8	79.4
	CCR+LG+VA	13.5	20.7	35.1	43.3	57.8	79.4
1	DM	12.4	15.5	28.9	39.2	59.8	78.4
	DM+LG	16.5	20.7	31.0	38.2	64.0	<b>81.5</b>
	CCR	15.5	18.6	31.0	45.4	59.8	78.4
	CCR+LG	<b>22.7</b>	<b>22.7</b>	<b>41.3</b>	<b>51.6</b>	<b>66.0</b>	80.5
	CCR+LG+VA	19.6	21.7	39.2	49.5	62.9	78.4

( DM: direct matching of raw SIFT features. CCR: compressive change retrieval of BoW. LG: local geometry. VA: visibility analysis. )

TABLE II  
PERFORMANCE GAIN

#ref	algorithm	top-1	top-2	top-5	top-10	top-20	top-50
40	DM	0.0	0.0	<b>0.0</b>	<b>0.0</b>	0.0	0.0
	DM+LG	0.0	0.0	<b>0.0</b>	<b>0.0</b>	0.0	0.0
	CCR	<b>+1.1</b>	0.0	-8.3	-7.3	+1.1	0.0
	CCR+LG	-4.2	<b>+3.1</b>	<b>0.0</b>	-3.1	<b>+7.3</b>	<b>+7.3</b>
	CCR+LG+VA	-5.2	+2.1	-3.1	-8.3	+4.2	+2.1
20	DM	0.0	0.0	0.0	0.0	0.0	0.0
	DM+LG	<b>+7.3</b>	<b>+6.2</b>	<b>+2.1</b>	<b>+7.3</b>	+3.1	<b>+2.1</b>
	CCR	0.0	+3.1	<b>+2.1</b>	+4.2	<b>+4.2</b>	-1.1
	CCR+LG	-1.1	+1.1	-1.1	+2.1	-1.1	-2.1
	CCR+LG+VA	+2.1	+4.2	<b>+2.1</b>	+3.1	-1.1	-1.1
10	DM	<b>0.0</b>	0.0	0.0	0.0	0.0	0.0
	DM+LG	<b>0.0</b>	<b>+2.1</b>	<b>+4.2</b>	<b>+2.1</b>	-3.1	+1.1
	CCR	-7.3	-3.1	+3.1	+1.1	<b>+1.1</b>	<b>+2.1</b>
	CCR+LG	-6.2	-7.3	-1.1	-4.2	-6.2	+1.1
	CCR+LG+VA	-5.2	-6.2	-3.1	-6.2	-8.3	0.0
5	DM	<b>0.0</b>	<b>0.0</b>	0.0	0.0	0.0	0.0
	DM+LG	-6.2	-2.1	<b>+5.2</b>	<b>+6.2</b>	+1.1	+2.1
	CCR	-4.2	-3.1	+3.1	-2.1	<b>+4.2</b>	<b>+5.2</b>
	CCR+LG	-3.1	-1.1	+3.1	-1.1	-3.1	+3.1
	CCR+LG+VA	-5.2	-1.1	+4.2	-3.1	-4.2	+3.1
1	DM	0.0	0.0	0.0	0.0	0.0	0.0
	DM+LG	+4.2	+5.2	+2.1	-1.1	+4.2	<b>+3.1</b>
	CCR	+3.1	+3.1	+2.1	+6.2	0.0	0.0
	CCR+LG	<b>+10.4</b>	<b>+7.3</b>	<b>+12.4</b>	<b>+12.4</b>	<b>+6.2</b>	+2.1
	CCR+LG+VA	+7.3	+6.2	+10.4	+10.4	+3.1	0.0